|  |  |
| --- | --- |
| 一般模式： 移动光标的方法 | |
| h 或 向左方向键(←) | 光标向左移动一个字符 |
| j 或 向下方向键(↓) | 光标向下移动一个字符 |
| k 或 向上方向键(↑) | 光标向上移动一个字符 |
| l 或 向右方向键(→) | 光标向右移动一个字符 |
| 如果想要进行多次移动的话，例如向下移动 30 行，可以使用 "30j" 或 "30↓" 的组合按键， 亦即加上想要进行的次数(数字)后，按下动作即可！ | |
| [Ctrl] + [f] | 屏幕『向下』移动一页，相当于 [Page Down]按键 (常用) |
| [Ctrl] + [b] | 屏幕『向上』移动一页，相当于 [Page Up] 按键 (常用) |
| [Ctrl] + [d] | 屏幕『向下』移动半页 |
| [Ctrl] + [u] | 屏幕『向上』移动半页 |
| + | 光标移动到非空格符的下一列 |
| - | 光标移动到非空格符的上一列 |
| n<space> | 那个 n 表示『数字』，例如 20 。按下数字后再按空格键，光标会向右移动这一行的 n 个字符。例如 20<space> 则光标会向后面移动 20 个字符距离。 |
| 0 | 这是数字『 0 』：移动到这一行的最前面字符处 (常用) |
| $ | 移动到这一行的最后面字符处(常用) |
| H | 光标移动到这个屏幕的最上方那一行 |
| M | 光标移动到这个屏幕的中央那一行 |
| L | 光标移动到这个屏幕的最下方那一行 |
| G | 移动到这个文件的最后一行(常用) |
| nG | n 为数字。移动到这个文件的第 n 行。例如 20G 则会移动到这个文件的第 20 行(可配合 :set nu) |
| gg | 移动到这个文件的第一行，相当于 1G 啊！ (常用) |
| n<Enter> | n 为数字。光标向下移动 n 行(常用) |
| 一般模式： 查找与替换 | |
| /word | 向光标之下寻找一个字符串名称为 word 的字符串。例如要在文件内查找 vbird 这个字符串，就输入 /vbird 即可！ (常用) |
| ?word | 向光标之上寻找一个字符串名称为 word 的字符串。 |
| n | 这个 n 是英文按键。代表『重复前一个查找的动作』的意思。举例来说， 如果刚刚我们执行 /vbird 去向下查找 vbird 这个字符串，则按下 n 后，会向下继续查找下一个名称为 vbird 的字符串。如果是执行 ?vbird 的话，那么按下 n 则会向上继续查找名称为 vbird 的字符串！ |
| N | 这个 N 是英文按键。与 n 刚好相反，为『反向』进行前一个查找动作。 例如 /vbird 后，按下 N 则表示『向上』查找 vbird 。 |
| :n1,n2s/word1/word2/g | n1 与 n2 为数字。在第 n1 与 n2 行之间寻找 word1 这个字符串，并将该字符串替换为 word2 ！举例来说，在 100 到 200 行之间查找 vbird 并替换为 VBIRD 则： 『:100,200s/vbird/VBIRD/g』。(常用) |
| :1,$s/word1/word2/g | 从第一行到最后一行寻找 word1 字符串，并将该字符串替换为 word2 ！(常用) |
| :1,$s/word1/word2/gc | 从第一行到最后一行寻找 word1 字符串，并将该字符串替换为 word2 ！且在替换前显示提示字符给使用者确认 (conform) 是否需要替换！(常用) |
| 一般模式： 删除、复制与粘贴 | |
| x, X | 在一行字当中，x 为向后删除一个字符 (相当于 [del] 按键)， X 为向前删除一个字符(相当于 [backspace] 亦即是退格键) (常用) |
| nx | n 为数字，连续向后删除 n 个字符。举例来说，我要连续删除 10 个字符， 『10x』。 |
| dd | 删除游标所在的那一整行(常用) |
| ndd | n 为数字。删除光标所在的向下 n 行，例如 20dd 则是删除 20 行 (常用) |
| d1G | 删除光标所在到第一行的所有数据 |
| dG | 删除光标所在到最后一行的所有数据 |
| d$ | 删除游标所在处，到该行的最后一个字符 |
| d0 | 那个是数字的 0 ，删除游标所在处，到该行的最前面一个字符 |
| yy | 复制游标所在的那一行(常用) |
| nyy | n 为数字。复制光标所在的向下 n 行，例如 20yy 则是复制 20 行(常用) |
| y1G | 复制光标所在列到第一列的所有数据 |
| yG | 复制光标所在列到最后一列的所有数据 |
| y0 | 复制光标所在的那个字符到该行行首的所有数据 |
| y$ | 复制光标所在的那个字符到该行行尾的所有数据 |
| p, P | p 为将已复制的数据在光标下一行粘贴，P 则为贴在游标上一行！ 举例来说，我目前光标在第 20 行，且已经复制了 10 行数据。则按下 p 后， 那 10 行数据会贴在原本的 20 行之后，亦即由 21 行开始贴。但如果是按下 P 呢？ 那么原本的第 20 行会被推到变成 30 行。 (常用) |
| J | 将光标所在列与下一列的数据结合成同一列 |
| c | 重复删除多个数据，例如向下删除 10 行，[ 10cj ] |
| u | 复原前一个动作。(常用) |
| [Ctrl]+r | 重做上一个动作。(常用) |
| . | 小数点！意思是重复前一个动作的意思。 如果您想要重复删除、重复粘贴等等动作，按下小数点『.』就好了！ (常用) |
| 进入编辑模式 | |
| i, I | 插入：在目前的光标所在处插入输入之文字，已存在的文字会向后退； 其中， i 为『从目前光标所在处插入』， I 为『在目前所在行的第一个非空格符处开始插入』。 (常用) |
| a, A | a 为『从目前光标所在的下一个字符处开始插入』， A 为『从光标所在行的最后一个字符处开始插入』。(常用) |
| o, O | 这是英文字母 o 的大小写。o 为『在目前光标所在的下一行处插入新的一行』； O 为在目前光标所在处的上一行插入新的一行！(常用) |
| r, R | 替换：r 会替换光标所在的那一个字符；R会一直替换光标所在的文字，直到按下 ESC 为止；(常用) |
| 上面这些按键中，在 vi 画面的左下角处会出现『--INSERT--』或『--REPLACE--』的字样。 由名称就知道该动作了吧！！特别注意的是，我们上面也提过了，你想要在文件里面输入字符时， 一定要在左下角处看到 INSERT/REPLACE 才能输入喔！ | |
| Esc | 退出编辑模式，回到一般模式中(常用) |
| 指令列命令模式 | |
| :w | 将编辑的数据写入硬盘文件中(常用) |
| :w! | 若文件属性为『只读』时，强制写入该文件。不过，到底能不能写入， 还是跟您对该文件的文件权限有关啊！ |
| :q | 离开 vi (常用) |
| :q! | 若曾修改过文件，又不想储存，使用 ! 为强制离开不储存文件。 |
| 注意一下啊，那个惊叹号 (!) 在 vi 当中，常常具有『强制』的意思～ | |
| :wq | 储存后离开，若为 :wq! 则为强制储存后离开 (常用) |
| :e! | 将文件还原到最原始的状态！ |
| ZZ | 若文件没有更动，则不储存离开，若文件已经经过更动，则储存后离开！ |
| :w [filename] | 将编辑的数据储存成另一个文件（类似另存新文件） |
| :r [filename] | 在编辑的数据中，读入另一个文件的数据。亦即将 『filename』 这个文件内容加到游标所在行后面 |
| :n1,n2 w [filename] | 将 n1 到 n2 的内容储存成 filename 这个文件。 |
| :! command | 暂时离开 vi 到指令列模式下执行 command 的显示结果！例如 『:! ls /home』即可在 vi 当中察看 /home 底下以 ls 输出的文件信息！ |
| :set nu | 显示行号，设定之后，会在每一行的前缀显示该行的行号 |
| :set nonu | 与 set nu 相反，为取消行号！ |